

REMARKS

Claims 1-22 are pending. Claims 1, 3, 8-10, 14, 15, 17-19, 21, 22 have been amended. Support for the amendments is found in paragraphs 29, 58 and 59, and FIG. 10. No claims have been added.

The applicant respectfully requests reconsideration of the application in view of the foregoing amendments and the following remarks, and respectfully requests that all rejections be withdrawn.

Rejection of Claims 1-22 under 35 U.S.C. § 102(e)

Claims 1-22 were rejected under U.S.C. § 102(e) as allegedly anticipated by U.S. Patent Application No. 2006/0085798 to Bendiksen et al. ("Bendiksen").

In rejecting claim 1, the examiner asserts that Bendiksen teaches:

for each process data item, identifying a corresponding process instance with which the process data item is associated, the corresponding process instance being a single execution of a corresponding process [Office action, page 3, lines 6-8]

The applicant respectfully disagrees.

The meaning of the term "process instance" has been clarified in the claims as amended. A process instance does not arise as an instance of a computational component. Instead, a process instance is discovered by the method of claim 1 from the application data found using predefined rules as multiple process components execute and multiple data items are collected.

The passage upon which the examiner relies reads as follows:

... group events automatically into related transactions, either within a single thread of execution and unit of work... [Bendiksen, paragraph 107]

Bendiksen further states:

In general, given a starting event (c) of interest to the user, the transaction analysis module can locate other events that occurred within the same local or business transaction as the event of interest. [Bendiksen, paragraph 108]

A business transaction in Bendiksen includes the operations done in one or more related local transactions.

In Bendiksen all captured events for a particular thread of execution are stored in a database. Bendiksen describes in paragraphs 58 – 63 and illustrates it in FIG. 3 a data model for storing and representing all captured events. According to this description, the program instance or thread of execution is a characteristic of the captured events. Hence, events stored in the database are inherently distinctive and distinguished by thread of execution. Among other things, the event data includes the name of the method/API call and time stamps.

Bendiksen teaches in paragraph 62 that each event entry is associated with a group of event relationships, including the method path relation. In paragraph 63 Bendiksen teaches a lookup table for storing mapping key-values with a set of events to group and lookup events in an efficient manner. The key-values include message ID, correlation ID and message time. The message path relation indicates a message transaction.

In Bendiksen, the transactions pre-exist and are identified in the captured events. The database model for storing the captured events is based on this notion. The central system knows at the outset to what thread of execution and to what method of a transaction each of the captured events pertains. This information is stored in the database. Furthermore, a lookup table is created to serve as a hash table for efficiently locating related events.

In the paragraphs cited by the examiner in rejecting claim 1, Bendiksen discloses a process for locating in the database a set of time sorted events of a particular transaction:

... the user specifies an event (e) of interest, and at step 912 the analyzer locates the event of interest in the time-sorted set of database 20 events, S, for event e's thread of execution. The resulting position in S is denoted as P. At step 914 the event at the current position in S is added to a set of events for the transaction. A test is then made at step 916 to determine if this event began the unit of work. If it did not, control passes to step 918 to find a previous event in S, ...[Bendiksen, paragraph 111]

Accordingly, the events from the transaction of interest are located by scrolling the stored events belonging to a particular thread of execution and checking the transaction identification

data, i.e., time stamps, method calls message ID, correlation ID, etc. The examiner will note that none of this data is application data.

Claim 1 as amended recites comparing each received process data item with one or more other received process data items to identify common application data. A plurality of process data items having common application data are grouped and a process instance associated with the group is discovered. The relationship between the process data items is identified not by the pre-existing notion of a thread of execution or instance of a computational component, but by the common application data that was found in the data items. Bendiksen in a sense describes operations for locating a transaction, which is the opposite of what claim 1 is directed to.

Further, claim 1 recites generating a reconstruction of a process instance. The examiner asserts that Bendiksen discloses the same functionality at paragraphs 122 and 123. The applicant respectfully disagrees.

In the cited paragraphs Bendiksen teaches transaction visualization in a user interface. The visualization in Bendiksen provides a user with an overview of a known transaction or set of transactions by arranging the collected information. The reconstruction recited in the claim, on the other hand, is a reconstruction based on the common application data found in the received process data items. This is an abstraction which can provide insight about a process without a pre-existing notion of the process or of its execution instances or threads of execution.

For each of the foregoing reasons, the applicant respectfully submits that claim 1 as amended is in condition for allowance.

Claim 10 as amended is directed to an agent collecting process data items and to transferring them to a central system operable to discover and reconstruct a process instance based on the process data items. The examiner asserts that Bendiksen discloses an agent providing the same functionality in paragraphs 47, 50, 51, 53-55, 60, 122 and 123.

The applicant respectfully disagrees.

Bendiksen discloses an analyzer system for monitoring a computing environment:

FIG. 1 illustrates an exemplary analyzer monitoring environment. An analyzer system 10 in accordance with the teachings herein comprises two major sub-systems: an analyzer 12 (also referred to herein as an analyzer

console) and a plurality of sensors 14. The sensors 14 may be considered as agents that reside in the space of a monitored process, and operate to collect information on calls of the particular technology that a particular sensor 14 is monitoring. [Bendiksen, paragraph 47]

In rejecting claim 10, the examiner suggests that the second subsystem of Bendiksen, i.e., the plurality of sensors, meets the limitations of claim 10. However, in paragraph 47, quoted above, Bendiksen clearly states that multiple sensors are required to collect the information for analysis. Then, this information is transferred to the central system, or the analyzer, for monitoring. Claim 10 as amended recites that the process data items collected by a single agent are sufficient for the central system to discover and reconstruct a process instance.

Further, in the arguments made above in reference to the rejection of claim 1, the applicant explained the meaning of reconstruction and process instance in the claims, and how that differs from what is described in Bendiksen. For those reasons, too, the applicant respectfully submits that Bendiksen does not disclose a central system operable to discover and reconstruct a process instance as recited in the claims.

Claims 14, 15, and 19 have been amended to recite limitations corresponding to the limitations of claim 1 as amended. The applicant respectfully submits that claim 14 as amended and claim 15 as amended are in condition for allowance for at least the reasons set forth above in reference to claim 1.

Claims 18 and 22 have been amended to recite limitations corresponding to the limitations of claim 10 as amended. The applicant, therefore, respectfully submits that claims 18 and 22 are in condition of allowance.

Claims 2-9, 11-13, 16-17 and 20-21 depend from claims 1, 10, 15 and 19 respectively. In addition to any independent basis for patentability, these dependent claims are patentable by virtue of at least this dependency.

DRAWINGS

The applicant again respectfully requests that the examiner indicate that the drawing sheets, FIGS 1-10, filed on March 24, 2004, are acceptable.

INTERVIEW SUMMARY

The applicant thanks Examiner Zhen for his courtesy in filling in for examiner Cao for an in-person interview on January 24, 2008. Attending were Examiner Zhen and applicant's representative Hans Troesch. The Examiner Zhen's reading of claim 1 was discussed.

CONCLUSION

For the foregoing reasons, the applicant submits that all the claims are in condition for allowance.

By responding in the foregoing remarks only to particular positions taken by the examiner, the applicant does not acquiesce with other positions that have not been explicitly addressed. In addition, the applicant's arguments for the patentability of a claim should not be understood as implying that no other reasons for the patentability of that claim exist.

The fees in the amount of \$120.00 for one month extension of time is being paid concurrently herewith on the Electronic Filing System (EFS) by way of Deposit Account authorization. Please apply any other charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: 15 Feb 08

/Hans R. Troesch/

Hans R. Troesch
Reg. No. 36,950

Customer No.: 32864
Fish & Richardson P.C.
Telephone: (650) 839-5070
Facsimile: (650) 839-5071